Towards Understanding the Advertiser's Perspective of Smartphone User Privacy

Yan Wang[†], Yingying Chen[†], Fan Ye[‡], Jie Yang[§], Hongbo Liu[♯]
[†]Stevens Institute of Technology
Castle Point on Hudson, Hoboken, NJ 07030
{*ywang48, yingying.chen*}*@stevens.edu*
[‡]Stony Brook University, Stony Brook, NY 11794
fan.ye@stonybrook.edu
[§]Florida State University, Tallahassee, FL 32306
jyang5@fsu.edu
[♯]Indiana University-Purdue University Indianapolis, Indianapolis, IN 46202
hl45@iupui.edu

***Abstract*—Many smartphone apps routinely gather various private user data and send them to advertisers. Despite recent study on protection mechanisms and analysis on apps' behavior, the understanding about the consequences of such privacy losses remains limited. In this paper we investigate how much an advertiser can infer about users' social and community relationships by combining data from multiple applications and across many users. After one month's user study involving about 200 most popular Android apps, we find that an advertiser can infer 90% of the social relationships. We further propose a privacy leakage inference framework and use real mobility traces and Foursquare data to quantify the consequences of privacy leakage. We find that achieving 90% inference accuracy of the social and community relationships requires merely 3 weeks' user data. The discoveries underscore the importance of early adoption of privacy protection mechanisms.**

***Keywords*-smartphone; privacy; social relationship**

I. Introduction

The huge success of smartphones is largely fueled by the availability of millions of phone apps that provide functions covering all aspects of our lives. A large portion of these apps are free. Their developers get financial support from advertisers by embedding their advertisement libraries to display mobile advertisements to users. Many advertisers exist and some of the major players include Google, DoubleClick and AdMob [1], [2]. To gain better understanding of user habits and behaviors for accurate ad targeting, these apps customarily scavenge private user data, ranging from the phone's IMEI number, MAC addresses of nearby access points, the user's location, even the contact list, and send it to advertisers [3], [4]. Ultimately, these "free" apps are not entirely free: users pay the price of their privacy.

There has been quite some recent work that investigates the privacy leakage and potential defense mechanisms. TaintDroid [3] can track the flow of different kinds of private information (e.g., IMEI, location) within an app and log the leaking of such information through network interfaces. Barrera et al. [5] and Felt et al. [6] examined permissions requested by about 1,000 apps and found requests for unnecessary permissions commonly exist. A number of tools [7], [8] can help users manage permissions granted to apps such that they do not have access to certain private information. Sowayway [9] and Kirin [10] can detect over-privileged apps or identify requests of dangerous combinations of permissions. Agarwal et al. [11] proposed a crowdsourcing based mechanism to help users decide proper privacy settings for iOS apps.

In this paper, we seek to answer an important but different question: how much does the advertiser know about the user, in particular, her social and community relationship (e.g.,family, colleagues and friends) from the leaked private data? This is motivated by a couple observations. First, there is only limited study of apps' dynamic leakage behavior at run-time. Existing study [4]–[6], [9], [12]–[14] is mostly on the static aspects of apps' permissions. TaintDroid [3] and D2Taint [15] can be used to log the leaking activities but the papers did not focus on a systematic study on the destinations, frequencies and types of apps' run-time privacy leakages. Second, the *consequences* of such leakage, especially when an advertiser gathers such private data from many users and across many apps, is not known either. It is easy to conjecture that the advertiser may gain additional information when cross-examining private data, but exactly what can be learnt, remains an open issue.

We focus on one important aspect of that perspective, the social and community relationships of a user, such as her family, colleagues and friends. Such knowledge is an important channel for the advertiser to push relevant advertisements since people tend to take note on things their acquaintances have done (e.g., bought). For example Facebook has largely relied on people voluntarily publicizing such relationship. However, many real world relationships are not publicized online yet they are equally important to advertisers; and there is a trend for Facebook users of various age groups to go for other "small-circle" social networks, or become less and less active due to privacy concerns [16].

In particular, we quantify to what extent an advertiser can learn and infer users' relationships by developing a privacy leakage inference framework. Our systematic study on privacy leakage inference involves both real experiments with multiple volunteers as well as trace-driven studies with human mobility traces obtained from two data sets, namely MIT reality trace [17] and Foursquare trace [18]. By examining the privacy

This is the author's manuscript of the article published in final edited form as:

Wang, Y., Chen, Y., Ye, F., Yang, J., & Liu, H. (2015). Towards Understanding the Advertiser's Perspective of Smartphone User Privacy. In 2015 IEEE 35th International Conference on Distributed Computing Systems (ICDCS) (pp. 288–297). http://doi.org/10.1109/ICDCS.2015.37

leakages of participants from a diverse background ranging from academia to city environments (i.e., our real experiments and the MIT trace are academia whereas the Foursquare trace represents a city environment), we discover that the privacy leakage enables an advertiser to infer a significant portion of a user's real world relationships that have physical interactions.

Specifically, we make the following contributions:

- We conduct a manual study of the frequencies, destinations and types of the *run-time* privacy leakages of nearly 200 most popular apps across 19 categories in Google Play. We discover that major advertisers can easily gather all types of private data in short time from many users.
- We model the relationship inference process in a three-layer framework and define the concept of *connection*, which is exemplified by two users sharing similar patterns in their leaked data (e.g., common Wi-Fi access points). We conduct a one-month real experiment of 10 participants of family, colleague and friend relationships, using various apps in their daily lives. We find that by aggregating data across users and apps, an advertiser can infer over 90% of the relationships from the "connections".
- We further propose two models, Activeness Based Profile and Probability Based Profile, for users' temporal privacy leakage profiles based on the experimental study. To verify the generality of findings from the real experiments based on privacy leakage inference, we conduct trace-driven studies by populating the derived user profiles to the human mobility traces in the MIT reality [17] and the Foursquare datasets [18]. We find that the advertiser can infer 80-95% of a regular user's relation in academia and city environments after gathering only 3 weeks of private data.

The rest of the paper is organized as follows: we present an overview of our approach in Section II and describe the run-time privacy leakage study in Section III. We define the relationship inference framework and the connection concept, conduct experimental study, and propose the privacy leakage profiles in Section IV. In Section V, we conduct trace-driven evaluation of relationship inference using MIT and Foursquare datasets. We put our work into the context of the related work in Section VII. Finally, we conclude in Section VIII.

II. Approach Overview

To facilitate the understanding on the consequences of privacy leakages, we take a four-step approach: run-time privacy leakage study, privacy leakage inference framework construction, experimental study and profile modeling, and inference framework evaluation via trace-driven study, as depicted in Figure 1.

From the advertiser's perspective, we study two types of relations: *social relationship* and *social community*. The social relationship is defined as a pair-wise relationship between two users with certain kind of physical interactions such as colleagues, families, and friends (not virtual friends from online social networks). Whereas a social community involves more than two users, who usually appear at a location during the same time period for certain common interests. For example, a group of students taking the same class twice every week or

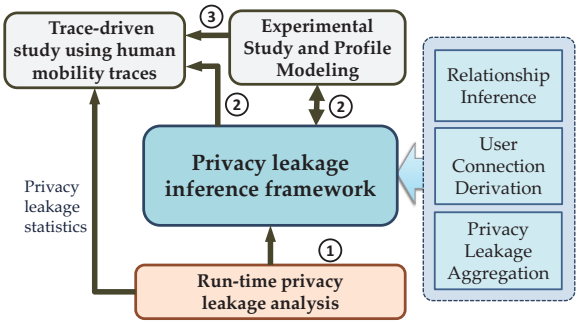


Fig. 1. We take a four-step approach to understand the advertiser's perspective on users' social and community relationships.

people eating in the same restaurant every Tuesday. We believe identification of both kinds of relationships help advertisers design better targeted advertising strategies.

Run-Time Privacy Leakage Study. First we want to systematically study the destinations, frequencies and types of the leaking behavior to understand the flow of common practice of privacy leakages in apps. We study their spatial and temporal privacy leakage characteristics to complement the existing work, which focuses on static aspects of permissions [7], [14], [19], or provides the capability of logging the leakage but stops short of a systematic study [3]. The Wall Street Journal (WSJ) study [20] in 2010 investigated the types of leakages for about 50 most popular apps, but not destinations and frequencies, and it was a bit outdated given the fast pace of the mobile market. Thus we conduct a series of experiments over a two-month period to obtain the most up-to-date picture of privacy leakages.

After manual testing of 190 most popular apps from 19 categories in Google Play (i.e., the largest Android application market), we find that major destinations commonly receive different types of private data from multiple apps, including geographical locations (GPS-location, network-based location), personal identities (phone number, IMEI), and communication information (contact list). We also find changes in the types of leakages for about half of the apps studied in [20].

Privacy Leakage Modeling. To understand the consequences of privacy leakages when an advertiser combines the data received from different users, we develop a three-layer privacy leakage inference framework (as depicted in Figure 1) including *Privacy Leakage Aggregation*, *User Connection Derivation* and *Relation Inference*.

We introduce an important concept *connection*, which exists when two users share similarities in leaked data. The *connection* helps bridge the gap between raw privacy leakage data and higher level relationship inference. The intuition is that each type of particular relationship has certain temporal-spatial patterns in users' physical interactions, which can be captured by *connection*. For example, two family members usually stay together at home during late night and early morning; while classmates encounter each other frequently in classrooms during the daytime of weekdays. Although exceptions to such patterns exist, it can usually identify most relationships and is the standard practice widely adopted in social community inference [21], [22].

We further conduct an experimental study of 10 participants for over one-month time period. Their privacy leakages are captured and analyzed. We find that an advertiser can infer over 90% of the pairwise relationships by using *connections* (e.g., IMEI and GPS location). Furthermore, we observe that the temporal-spatial similarities between people who have friend relationship is not as regular as that between people of other relationships that have repetitive interactions (e.g., colleagues and families).

Evaluation of User Privacy Inference. To understand whether the above observations can be generalized to larger scale user population with various backgrounds, we extract user privacy leakage profiles, apply them to user mobility traces generated from two datasets with over 500 participants. We verify that using 3 weeks of private data, an advertiser can infer colleague-based relationships of regular users at around 90% accuracy in an academia environment, and friend-based relationships above 95% in a city environment. When an advertiser uses hierarchical clustering to infer social communities, $80-90\%$ of those of regular users' are revealed in academia environments, and over 80% in a city environment.

III. Run-Time Privacy Leakage Study

Current smartphone operating systems only have coarse-grained control on whether an application can access users' private data [13]. For example, Android controls the access through the install-time permission system [23]. Once the access is granted, the application is free to access it as frequently as possible and send it to wherever it wants over the Internet [1], [3], [9].

The most similar study to ours is the WSJ one [20] in 2010, which focuses on 5 privacy leakage types (i.e., *contacts, location, phone id,* and *phone number*) among 50 most popular apps from Google Play. The results are interesting but there is no analysis on the frequencies and destinations of privacy leakages. Our study aims to provide a more comprehensive and up-to-date analysis including the frequencies and destinations of the leakages. We also investigate how much private data an advertiser can collect and aggregate from multiple apps. This helps the user understand the scope and extent of privacy leakages when running apps; it also serves as the basis for the formulation of the privacy leakage inference in the next section.

A. Methodology

We choose the top 10 most popular applications from each of the 19 categories in Google Play as of January 2013, totaling 190 applications. We expect that these most popular apps are installed by the majority of users, thus their behavior analysis is representative to the majority of users. To analyze the apps' behavior, we developed a tool leveraging certain capabilities of TaintDroid, which helps to track and log the privacy leakages of the applications. Some of the apps crash during the test, and we are able to gather complete results for 145 apps. We use two types of Android phones, Google Nexus One and Google Nexus S. To capture the application's behavior at different times in a day, we test about 4-5 apps in three different time periods (i.e., morning, noon and night) in each day. During each period, the selected apps are tested one at a time (i.e., we reboot and uninstall each application after testing it) for about 5 minutes. This procedure helps to avoid interference between applications.

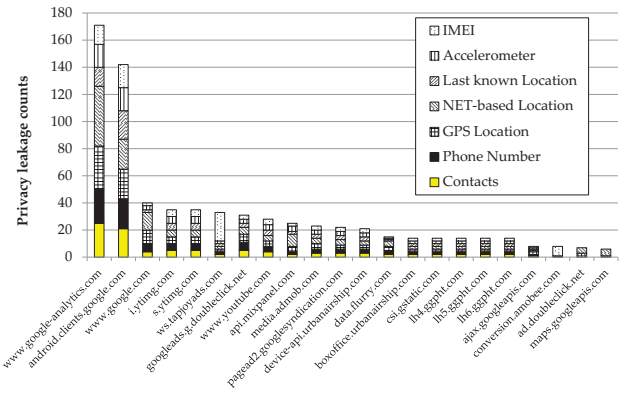


Fig. 2. Privacy leakage counts of destinations that collect private data from more than one app during one-day testing.

B. Findings

Per App Privacy Leakage. During our study, we find that about 50% of the apps in the WSJ report have changed their privacy leakage behavior. In particular, we find that 97 out of the 145 applications send out private data of the user. The data includes *GPS location, network-based location* (provided by Android based on cellular ID and Wi-Fi networks), *WiFi Access Point SSID list, contact list, phone number, International Mobile Station Equipment Identity (IMEI), accelerometer readings*, but not those of microphone, camera and text message log, which are accessed by the applications but not leaked out. We also find that there are 8 applications sending Wi-Fi SSID list (scanned by the smartphone) through SSL, and 3 of the 8 applications (i.e., Compass, CNN App, Yelp) send this information to the same destination (with IP address 173.194.73.104 belonging to Google according to www.iplocation.net). This type of WiFi AP list is most likely the company's effort to build a WiFi address database for geo-location purposes [24]. It could be employed to infer the user's location, thus potentially her mobility pattern during a day.

Per Destination Privacy Leakage. We further investigate how the private data could be collected by a single destination (e.g., an advertiser's server) through multiple apps. We identify 22 "common" destinations(i.e., the destinations receiving data from more than one app) in 19 categories. Specifically, we find that three Google destinations collect 7 types of private data from more than 30 applications. Figure 2 shows that during one day's testing of the 97 applications, most of the 22 destinations collect more than 3 types of private data, and Google is the most active one among these destinations.

Privacy Leakage Frequency. During our testing, we find that the Location and IMEI are the first and second most common privacy leakage types, which involves 71 and 61 applications respectively. We present the leakage count decomposition of 28 apps that leaks more than 10 times in its 5-minute usage in Figure 3. We observe that 7 apps have leaked more than 4 types of private information, 3 of them have even leaked all 7 types of private information (i.e., Pandora Jewelry, Carphone Mobile Superhero, and Evernote). These results indicate that an advertiser has a comprehensive view

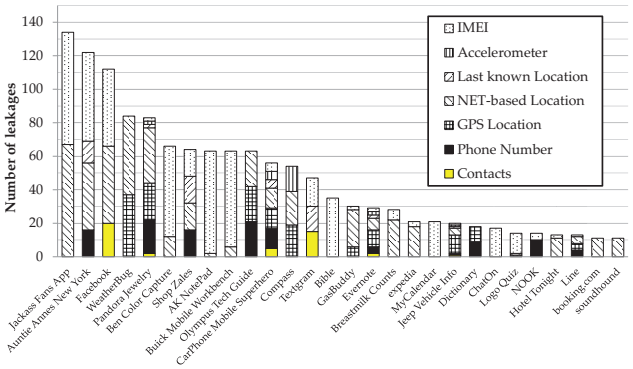


Fig. 3. Privacy leakage counts of apps leaking more than 10 times in a 5-minute continuous testing.

of the users' private data through their daily phone usage. Thus it can potentially use the combination of private data (such as IMEI and NET-based location) to identify the location of the user from such apps, and further obtain a fine-grained picture of the user's social life with the assistance of the leaked contact information.

Leakage to Common Destinations. We take a closer look at the leakage frequency from different apps to a "common" destination. We observe that WeatherBug and Jackass Fans are the top two apps with the most frequent leakages: they leak about 80/70 times during the 5 minute period. We further summarize the leakage frequency in app categories to common destinations and observe that the Weather category exhibits the highest privacy leakage frequency, partly due to their needs to know the user's location, and the Social category is the second. This again confirms that various apps leak private information to multiple common destinations, which allows the advertiser to piece together the user's social picture at fine temporal granularity through multiple apps.

IV. User Privacy Leakage Modeling and Experimental Study

The run-time privacy leakage study provides a comprehensive analysis on how much private data and advertiser can collect and aggregate from multiple apps, which serves as the basis for the formulation of the privacy leakage inference from multiple apps and across users. In this section, we present a privacy leakage inference framework that *quantifies to what extent an advertiser can learn and infer users' relationships*. We then run real experiments with multiple participants to analyze the consequences of the privacy leakage from the advertiser's perspective and abstract privacy leakage user profiles based on the experiments.

A. Privacy Leakage Modeling

We first define the concept of *connection*. A *connection* between two users exists if the same type of privacy leakage from the two users share certain spatial, temporal or content similarities. A few examples are:

Contact list: A connection instance exists between two users if they are in each other's contact list, or they share common contacts. (However, we note that contact lists are not sufficient for social relationship inference simply because a person does not necessarily have close relationship with everyone in his or her contact list.)

Wi-Fi Access Point list: A connection instance exists between two users when they share common leaked access points at the same time.

GPS location: A connection instance exists between two users if two GPS locations leaked around the same time are close by within a certain threshold.

Network-based location: A connection instance exists between two users if the leaked network-based locations are close by within a certain threshold around the same time.

The *connection* bridges the gap between the privacy leakage information and the users' relationship inference. In particular, to quantify the consequences of the privacy leakage from the advertiser's perspective, we design a privacy leakage inference framework, which consists of three virtual layers: *Privacy Leakage Aggregation*, *User Connection Derivation*, and *Relationship Inference* as shown in Figure 4.

Such a framework facilitates us to perform a systematic study to understand the advertiser's perspective of user privacy: (1) The Privacy Leakage Aggregation layer deals with the raw privacy leakage information. An advertiser can combine the privacy leakage data from multiple apps across different users over time. For example, the users can be identified by the IMEI or phone number. The aggregated privacy leakage data of each user can then be categorized into different types, such as contact list, AP list, GPS location, and Network location. (2) In the User Connection Derivation layer, the advertiser correlates the data from different users and identifies connections [1] between any two users. By correlating different types of privacy leakages across the users over time, the connection frequency between any two users can be derived. (3) In the Relationship Inference layer, the user's social and community relationships, such as family, colleagues and friends, are inferred based on the connections between users. The type of relationship is usually determined by examining the temporal and spacial patterns of the connections (e.g., family members usually have connections at home in the morning and at night, whereas colleagues have connections in office during working hours). We next conduct an experiment to study the effectiveness of our privacy leakage model using this framework.

B. Experimental Study

1) Design of Experiments: Our experiment involves 10 volunteer students and their family members over one month period, among which five types of relationships exist: colleague, collaborator, classmate, friend, and family. To clarify, collaborators are usually colleagues that actively work together, usually at regular times such as weekly meetings. We developed a tool to capture the privacy leakage information in real-time leveraging TaintDroid. During the experiments, we distribute smartphones to volunteers with the tool and the top 10 popular apps (across 19 categories in Google Play) installed. Because the experimental smartphones are not replacements of the volunteers' regular phones, they are asked to use their experimental smartphones at least three times a

[1] We use "connections" to refer to connection instances later in the paper.

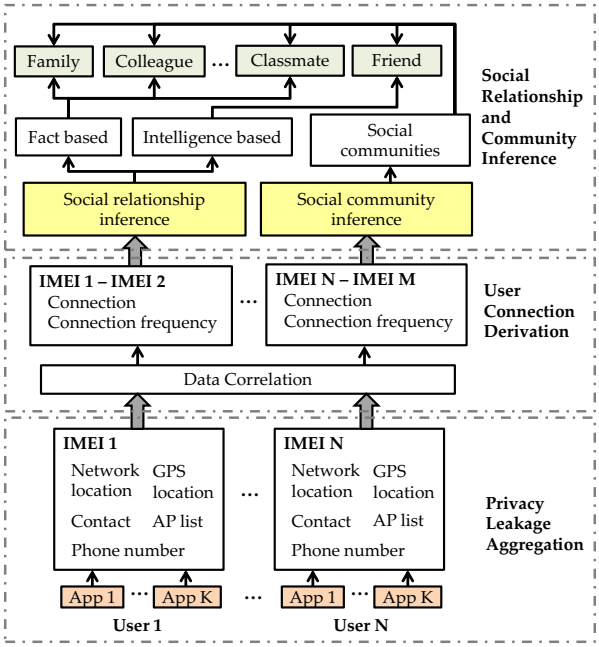


Fig. 4. Privacy leakage inference framework with three virtual layers to quantify the advertiser's perspective of smartphone user privacy.

day. The volunteers are also encouraged to use whichever apps they are interested in without knowing the purpose of this experiment. After the experiments we extract the leaked privacy data logged by our tool to quantify to what extent an advertiser can infer a user's relationships.

2) Observations: ***User Connection Derivation.*** Figure 5 shows one example on the temporal patterns of the derived user connections at a residential area based on GPS location leakage. In particular, subject1 and subject2 have frequent connections in the morning (around 10AM) and at night (through 9PM-2AM) at a residential area. The advertiser can thus infer the two subjects most likely have a family relationship. Additionally, the advertiser can also infer subjects' other social relationship, such as colleagues, collaborators, and friends, based on the spatial and temporal patterns of connections extracted from privacy leakages over time.

Definition of Two Types of Social Relationships. From the experimental results, we observe that while some relationships (e.g., family, colleagues, collaborators, and classmates) exhibit repetitive connection patterns, some others like friends do not. This is because family and colleague based relationships naturally carry similar spatial-temporal patterns dictated by the relationship. For example, families live together at night while colleagues work together during the day, whereas friendship does not necessarily carry such inherent patterns. Two friends that do not hang out for a while are still friends. We distinguish these two categories of relationships as *Fact Based Relationship* and *Intelligence Based Relationship*, which covers traditional social relationships. The *Fact-based Relationship* includes colleagues, classmates, roommates, families that carry inherit similar, regular and repetitive spatial-temporal connection patterns as dictated by the relationship, whereas the *Intelligence-based Relationship* includes friends, which do not necessarily carry such patterns.

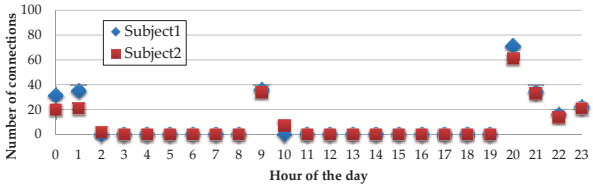


Fig. 5. Example of location leakage patterns of family relationship.

C. Making Inference based on Thresholding

We next investigate how accurate an advertiser could infer about the user's social relationship such as colleagues, families and friends by utilizing connections between users derived from different types of privacy leakages.

We use a *threshold-based* approach to derive relationships based on the connections between users extracted from privacy leakage aggregation at the advertisement provider. If the connection count between two users exceeds a certain threshold in an observation window, we consider some relationship exists. Our framework utilizes the temporal and spatial patterns of the connections to classify the type of relationship: the connections of colleagues occur in work hours of weekdays, families in early morning and late night, while friends after working time and in weekends.

These simple rules may not be entirely reliable. Nevertheless, we show that an advertiser can make inference even with such simple rules. We note that such temporal and spatial patterns of the connections are the basis to statistically generate a user's privacy leakage profile, which will be described in the next subsection.

In total we have 10 pairs of colleague relationship, 5 pairs of collaborator relationship, 1 pair of family relationship, 2 pairs of classmates, and 3 pairs of friend relationship among our 10 participants. That is 18 pairs of fact-based relationships and 3 pairs of intelligence-based relationships. During our experiments, we observe that by utilizing connection frequencies and patterns, an advertiser can infer over 90% social relationship correctly.

D. Deriving Privacy Leakage User Profiles

Based on the experimental data collected over one month, we next build the privacy leakage user profile to statistically capture the temporal and spatial patterns. We develop two types of privacy leakage user profile, *activeness based profile* and *probability based profile*, which will be applied to our large-scale trace-driven studies on advertiser's perspective in the next section.

1) Activeness Based Profile: The activeness based profiles are generated based on privacy leakages from each participant in our experiments and aim to capture the fine-grained statistical view of the privacy leakages.

Step 1. We first derive the privacy leakage probability model of a particular user. Assume there are N types of privacy leakages observed in total. We divide the time in day d into T time windows as $\{w_t, t = 1, \cdots, T\}$. Then within a time window w_t, a vector $\Phi^{u,d,t}$ is defined to capture the numbers of occurrences of different privacy leakage types, and each

element $\Phi^{u,d,t}(i)(i = 1, \cdots, N)$ corresponds to the number of times privacy leakage type i occurs. For example, when $\Phi^{u,d,t}$ equals to $[2, 1, 0]$, it means 2 occurrences of leakage type 1, 1 occurrence of leakage type 2 and 0 occurrence of leakage type 3 in time window W_t at day d for user u.

We then define $\gamma_i^{u,d,t}$ to indicate whether the privacy leakage type i appears in the vector $\Phi^{u,d,t}$ as:

$$\gamma_i^{u,d,t} = \begin{cases} 1, \Phi^{u,d,t}(i) \neq 0 \\ 0, \Phi^{u,d,t}(i) = 0. \end{cases} \quad (1)$$

The probability that type i leakage happens for user u in time window w_t across D days (e.g., $D = 7$ days) is defined as:

$$Prob_i^{u,t} = \frac{\sum_{d=1}^{D} \gamma_i^{u,d,t}}{D}. \quad (2)$$

Step 2. The number of occurrence of the privacy leakages affects the inference of a user's social community. Thus we capture the frequency of type i privacy leakage using the average number of occurrences over the days it happens in time window w_t across D days. Specifically, the average rate $r_i^{u,t}$ is defined as:

$$r_i^{u,t} = \frac{\sum_{d=1}^{D} \Phi^{u,d,t}(i)}{\sum_{d=1}^{D} \gamma_i^{u,d,t}}. \quad (3)$$

The activeness based profile of user u consists of $Prob_i^{u,t}$ and $r_i^{u,t}$.

Example. We illustrate the generation of the activeness based profile of user u in Figure 6. We examine a privacy leakage dataset across 7 days (i.e., one week) with the time window w_t set to 5 minutes and 288 time windows in total per day. Assume 3 types of privacy leakage are under study. For w_t at day 2, if there are 2 occurrences of leakage type 2 and 8 occurrences of type 3 observed, we have $\Phi^{u,2,t} = [0, 2, 8]$ and $\gamma_2^{u,2,t} = 1$ shown as the green eclipse in Figure 6. In addition, if the leakage type 2 is only observed during day 1 and day 2 with $\Phi^{u,1,t} = [0, 5, 7]$ and $\Phi^{u,2,t} = [0, 2, 8]$, the privacy leakage probability of type 2 privacy leakage in time window w_t across 7 days for user u can be calculated using Equation (2) as: $Prob_2^{u,t} = \frac{\gamma_2^{u,1,t}+\gamma_2^{u,2,t}+\cdots+\gamma_2^{u,7,t}}{7} = \frac{1+1+\cdots+0}{7} = 0.286$. And the corresponding average rate can be obtained as: $r_2^{u,t} = \frac{\Phi_2^{u,1}+\Phi_2^{u,2}+\cdots+\Phi_2^{u,7}}{\gamma_2^{u,1,t}+\gamma_2^{u,2,t}+\cdots+\gamma_2^{u,7,t}} = \frac{5+2+\cdots+0}{1+1+\cdots+0} = 3.5$, which is shown in blue rectangles in Figure 6.

Categorization. Once the activeness based user profile is obtained, the advertiser could further categorize the profiles by the number of hours k_u the user u has privacy leakages in a one-day duration. There are three representative user categories, namely *active user category*, *regular user category*, and *inactive user category*. Assume two thresholding hours ρ_1 and ρ_2 with $\rho_1 > \rho_2$. If the user u has greater than ρ_1 hours with privacy leakages, his user profile is put into the active user category. If the user u has less than ρ_1 but larger than or equal to ρ_2 hours with privacy leakages, his user profile is then added into the regular user category. When the user u has less than ρ_2 hours privacy leakages, his user profile is then captured in the inactive user category. The categorization can be summarized as:

$$\alpha = \begin{cases} 1 \text{ (active user category), if } k_u \geqslant \rho_1; \\ 2 \text{ (regular user category), if } \rho_2 \leqslant k_u < \rho_1; \\ 3 \text{ (inactive user category), if } k_u < \rho_2. \end{cases} \quad (4)$$

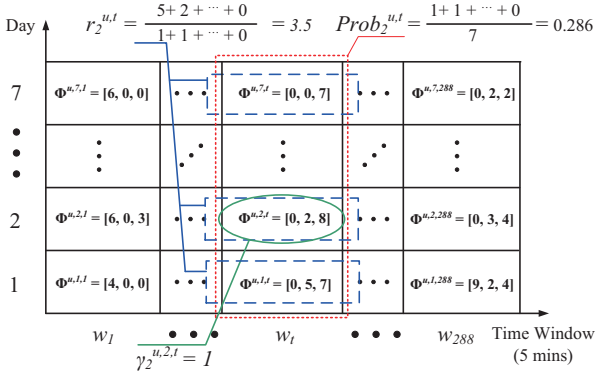


Fig. 6. Example of activeness based profile generation for user u with 3 types of privacy leakages when $D = 7$.

2) Probability Based Profile: The $\{Prob_i^{u,t}, r_i^{u,t}\}$ in each activeness based profile captures the leakage of the corresponding user u. To characterize the statistical average of the leakages of users in the same category, we design the probability based profile. Activeness based profiles in the same category are used to derive the leakage probability and rate for the probability based profile of that category.

Step 1. We first define the $\delta_i^{u,t}$ to indicate whether there is a probability of type i leakage in the time window w_t for the profile of user u as:

$$\delta_i^{u,t} = \begin{cases} 1, Prob_i^{u,t} \neq 0 \\ 0, Prob_i^{u,t} = 0. \end{cases} \quad (5)$$

Then we calculate the average probability that leakage type i occurs in a time window for user u:

$$\lambda_i^u = \frac{\sum_{t=1}^{T} \delta_i^{u,t}}{T}. \quad (6)$$

Step 2. We then define the leakage probability of that category as the previous probability averaged over all users of the same category.

$$Prob_i^{\alpha} = \frac{\sum_{u=1}^{M_\alpha} \lambda_i^u}{M_\alpha}, \quad (7)$$

where M_α is the number of users belonging to the category α.

Step 3. The corresponding profile privacy leakage rate is calculated over all the users in one particular category α as:

$$r_i^{\alpha} = \frac{\sum_{u=1}^{M_\alpha} \sum_{t=1}^{T} r_i^{u,t}}{\sum_{u=1}^{M_\alpha} \sum_{t=1}^{T} \delta_i^{u,t}}. \quad (8)$$

The probability based profile of a user category α then consists of $Prob_i^{\alpha}$ and r_i^{α}. Based on the data from our experiments they can be calculated. We respectively name them as *high probability* ($Prob_i^{\alpha} = 0.87$), *medium probability* ($Prob_i^{\alpha} = 0.68$), and *low probability* ($Prob_i^{\alpha} = 0.44$) profiles.

Both types of profiles quantify the users' privacy leakage characteristics: the leakage probability $Prob_i^{u,t}, Prob_i^{\alpha}$ determine whether type i privacy leakage happens or not in a time window, whereas the average leakage rate $r_i^{u,t}, r_i^{\alpha}$ determine the number of type i leakages in that time window should they happen at all. The profiles will be used in our large-scale trace-drive studies in the next section to facilitate the understanding of the user relationship inference from the advertiser's point of view.

V. Social Relationship Inference Leveraging Privacy Leakages

In this section, we systematically study the consequence of the privacy leakages obtained by advertisers by applying the privacy leakage model to two human mobility traces. In particular, we study how much an advertiser can infer about users' social and community relationships by combining privacy leakages from multiple apps and across many users.

We build a simulator utilizing the privacy leakage inference framework to generate connections between users based on both the leakage profiles derived from the previous section and the human mobility traces. The human mobility traces are used to discover connections between users in both an academia and a city environment.

A. Methodology

1) Human mobility traces: We use two human mobility traces: the Foursquare trace [18], and the MIT trace [17], which come from different backgrounds and have various relationships. Specifically, the MIT trace represents participants with similar background in an academia environment, where user relationships mostly represent research colleagues, office staff and classmates. In the Foursquare trace, participants have more diverse relationships; they may be colleagues, friends, and families in a city environment. The details of these two traces are introduced below:

Foursquare Trace. Foursquare is a company that helps people share life experiences based on locations such as restaurants. This trace is generated based on tipping information collected from different venues in Los Angeles (LA). A tip in a venue shows that one participant has carried out some essential activities (like dinning and shopping) at that venue. There are $104,478$ tips left by $31,544$ participants in this trace. We choose 354 participants from the top 10 venues (which are all restaurants) to generate encounter events between participants based on the time of the tipping in a 21 day duration.

MIT Trace. This trace is collected on MIT campus for 10 months by 107 participants with smartphones. Each smartphone scans (using Bluetooth) and records nearby smartphones every five minutes. The encounter happens when two participants are located in close physical proximity (e.g., shown in the Bluetooth scanned neighboring list in MIT Trace). And such an event is defined as an *encounter event*. There are 97 participants with valid data including staffs and students. In our study, we use 21 days' data which includes 91 participants for social relationship inference.

2) Privacy Leakage Profile Population: To understand the impact of user profiles, we repeat the study using both activeness and probability based profiles. When activeness based ones are used, each participant is assigned a randomly selected profile in the chosen category (i.e., active, regular and inactive). When probability based ones are used, each participant is assigned the same probability profile (i.e., one of high, medium and low). When presenting our results, we will use terms like "active users" or "users of medium probability" to (loosely) refer to participants assigned of the activeness or probability based profiles.

We then infer relationship based on connections derived from the leakages over observation windows of different sizes (i.e., 7, 14 and 21 days). From real experiments with 10 participants having known relationships, we find that different thresholds of connection counts in the observation window should be applied to derive different relationships. We thus use 3 days for fact-based relationship and 2 days for intelligence-based relationship in both experiments and simulations. Such thresholds enable an advertiser to achieve over 90% inference accuracies for regular users with very small false positive rates, which is a good balance between the two.

Foursquare Trace. In order to apply privacy leakage profiles to this dataset, we generate encounter events between participants as follows: for a particular venue, we give a visiting duration with a random length ranging from 30 minutes to 2 hours to each tipping user. In the overlapped period of the duration of two users, we generate encounter events with a fixed time interval of 30 minutes (e.g., in an 1 hour overlapped period, we generate 2 encounter events). For each encounter event, we first find out corresponding 5-minute time windows. Then we flip a coin with the privacy leakage probability in the users' profile (defined in Equation (2) or (7)) to decide whether privacy leakages should happen or not in that 5-minute time window. If they do, we use the privacy leakage rate defined in Equation (3) or (8) as the number of leakages revealed to the advertiser in that particular 5-minute time window. The leakages are used by the advertiser to derive connections and eventually encounter events to infer users' relationship.

MIT Trace. The MIT trace records encounter events for each user every 5 minutes. Therefore, we use the same way as we introduced for the Foursquare trace to populate privacy leakages among users in the MIT trace based on users' encounter events.

B. Metrics

In our evaluations, we study the inference accuracy of pairwise social relationship and the correlation between social communities extracted based on the connections of users.

Inference accuracy. This is the ratio between the successfully inferred relationship pairs and all relationship pairs.

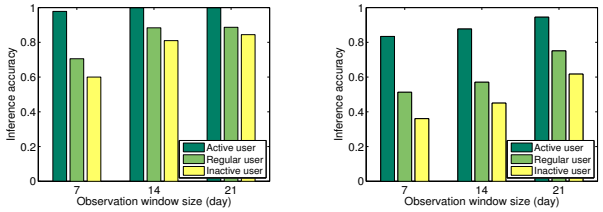
Community correlation. This is the ratio between the common subjects within a community identified by our privacy leakage inference framework and the total number of subjects within the community.

False positive rate. This is the ratio between the number of mistakenly identified members of inferred community and the total number of members within the inferred community.

C. Inference with Privacy Leakages

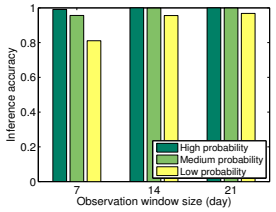
1) Combination of Privacy Leakages: As we discussed in the previous section, an advertiser can utilize the temporal and spatial patterns of connections to infer users' relationship. There are multiple privacy leakages that can produce connections between users. In this study we focus on the $\{$user identity, location$\}$ combinations of most popular privacy leakages including IMEI, phone number, GPS location, Wi-Fi AP list, and network-based location.

2) Pairwise Social Relationship Inference: Figure 7 compares the accuracy of pairwise social relationship inference (for both fact and intelligence based relationships) by applying

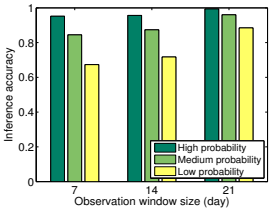


(a) Fact based relationship

(b) Intelligence based relationship

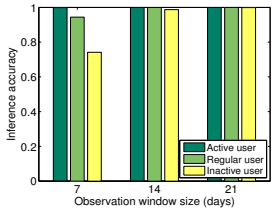


(c) Fact based relationship

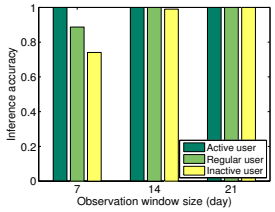


(d) Intelligence based relationship

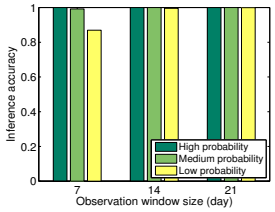
Fig. 7. Inference accuracy, MIT mobility trace, (a) and (b) are from the activeness based profiles, (c) and (d) are from the probability based profiles.



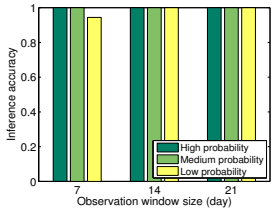
(a) Fact based relationship



(b) Intelligence based relationship



(c) Fact based relationship



(d) Intelligence based relationship

Fig. 8. Inference accuracy, Foursquare mobility trace, (a) and (b) are from the activeness based profiles, (c) and (d) are from the probability based profiles.

activeness and probability based profiles to the MIT trace under different sizes of observation windows. We use 3 days and 2 days as the threshold for fact-based and intelligence-based relationship inference respectively, which is introduced in the previous section (same threshold applies hereafter). The fact-based relationship has greater threshold because people having fact-based relationships are supposed to encounter each other more regularly than those having intelligence-based relationships (e.g., colleagues meet 3 days a week while friends meet 1 day a week).

From Figure 7 (a) and (b) we observe that for most cases, an advertiser can achieve over 80% inference accuracy for fact-based relationships with regular users, whereas it is around 60% for intelligence-based relationships. We also observe that the inference accuracy decreases for less active users, which is reasonable since less usage leads to less privacy leakages. In addition, for active users, Figure 7 shows that the inference accuracy for both fact-based relationship and intelligence relationship is high (i.e., above 90%).

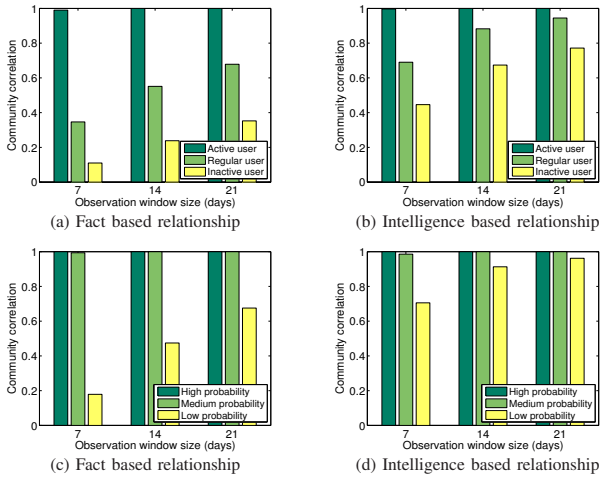
Furthermore, we find that longer observation windows help improve the inference accuracy, especially when the privacy leakage probability is low. This is observed in the probability-based approach shown in Figure 7 (c) and (d). It is because a longer window helps the advertiser to accumulate more data, resulting in more connections to identify users' relationships. Comparing Figure 7 (a) and (b) to Figure 7 (c) and (d) respectively, we observe that the inference accuracy of active users is similar to that of users have the profile with a high leakage probability. Figure 7 (d) suggests that in order to keep the inference accuracy of intelligence based relationship lower than 0.6, the user has to keep his leakage probability smaller than *low probability* (i.e., 0.44).

Examining the Foursquare trace, we observe much higher inference accuracy for both fact and intelligence based relationships. We show the results using privacy leakage profile with different activeness in Figure 8 (a) and (b). Even for inactive users, the inference accuracy is about 70% for a 7-day window, and it goes over 95% for the 14-day window. This is because users in the Foursquare trace encounter each other more frequently than those in the MIT trace. Thus, more connections can be discovered when the users have the same intensity of app usage, leading to a higher inference accuracy. This is also the case when using the probability-based profile to infer pairwise relationship with the Foursquare trace, as shown in Figure 8 (c) and (d).

3) Social Community Inference: We next study how the social community could be inferred by the advertiser using the privacy leakage profile. In particular, based on the inferred pairwise social relationships, a hierarchical clustering algorithm [25] is applied to obtain the social communities of users with similar relationships (e.g., collaborators, labmates, and classmates).

Community Correlation. We present the community correlations of both fact based relationship and intelligence based relationship when activeness and probability based profiles are applied to the Foursquare trace in Figure 9. We observe that the intelligence based community correlation is high for active users and regular users (i.e., over 80% on average). Similarly it is high for users with high probability and medium probability profiles (i.e., over 90%). However, the community correlation of users having the fact-based relationship is much lower with inactive users and users with low probability profiles (i.e., ranges from 10% to 60%). This is because participants in Foursquare data are from much diverse background in the city environment and the locations are mostly restaurants which favors more to the intelligence-based relationship inference. Furthermore, we observe that the community correlation also increase with longer observation windows, especially for regular users and inactive users. This is also because longer observation windows help the advertiser to aggregate more connections between users, which helps to more accurately identify their social communities. Since MIT trace has similar result, we do not provide figures for MIT trace due to the limited space.

Discussion of False Positive. Table I shows false positive rate of community correlation for both MIT and Foursquare



(a) Fact based relationship (b) Intelligence based relationship

(c) Fact based relationship (d) Intelligence based relationship

Fig. 9. Community correlation, Foursquare dataset, (a) and (b) are from the activeness based profiles, (c) and (d) are from the probability based profiles.

traces. Overall, the false positive rate is very small (i.e., the average is around 0.04). In addition, larger observation window results in smaller false positive rates. This is because longer observation window includes more information and thus can improve the inference performance. Furthermore, we observe that the Foursquare trace has much lower false positive rate than the MIT trace. This is because the Foursquare trace has more connections, thus leads to lower false positive rate.

To briefly summarize the major findings, 1) the advertiser can infer users' social relationships at high accuracy, e.g., over 90% on average for active users and over 80% on average for regular users; 2) the advertiser can also infer a significant portion of users' community relationships, e.g., over 90% on average for active users and over 60% on average for regular users, which reveals common interests or activities among users not necessarily with direct interactions.

VI. Discussion

More Types of Private Data. There are potentially other types of private data available to advertisers. For example, Google has the access to its search terms and histories of many users. Although we do not find the 190 apps studied leaking text message logs, audio and video data, illegitimate access and disclosure of such information are not impossible. Our current study is based on the combination of most basic privacy leakages (i.e., identities and locations). Contact list carries important information. For example, a certain relationship most likely exists when two users share many common contacts. Unfortunately, because we do not have the contact lists of the subjects of the two mobility traces, we are not able to evaluate its impact on relationship inference. Studying the impact of more types of private data would be interesting future work.

Large Scale Evaluation. We are aware that our experiments (due to limited available manpower) may cause bias to our privacy leakage profiles and the evaluation may not cover all privacy leakage patterns in the real world. Increasing the number of participants in real experiments and building more sophisticated privacy leakage profiles are our future exploration as well.

Insufficient Metadata. Our evaluation is constrained by the availability of metadata descriptions of datasets. Neither the MIT nor Foursquare data has sufficient annotation to differentiate the relationships among users at fine granularity desired by us: colleague, collaborator, classmate, friend, and family. In lieu of that, we have to utilize the most commonly used technique for detecting social communities (i.e., hierarchical clustering algorithm [25]) and use the results as the base of comparison, which is a common practice in social relationship research. In the future we hope such metadata would be made available when people conduct such experiments.

More Advanced Inference Algorithms. The thresholding algorithm that we use to infer relationships is based on quite simple heuristics. With the availability of large amount of data, advertisers can use more advanced inference algorithms, e.g., by utilizing data mining techniques. We are aware of this limitation and plan to build advanced models more closely describing general cases.

	Fact-based			Intelligence-based		
	7days	14days	21days	7days	14days	21days
MIT Trace						
Active users	0.035	0.022	0.016	0.149	0.129	0.098
Regular users	0	0	0	0.065	0	0
Inactive users	0	0	0	0.056	0	0
High prob.	0.069	0.054	0.052	0.17	0.145	0.129
Medium prob.	0.045	0.041	0.039	0.104	0.093	0.084
Low prob.	0.026	0.016	0.015	0.055	0.037	0.0257
Foursquare Trace						
Active users	0	0	0	0.019	0.076	0.076
Regular users	0	0	0	0.081	0.061	0.052
Inactive users	0	0	0	0.04	0.038	0.029
High prob.	0	0	0	0.19	0.078	0.078
Medium prob.	0	0	0	0.183	0.078	0.078
Low prob.	0	0	0	0.099	0.064	0.058

TABLE I
False positive rate for community correlation: MIT trace and Foursquare trace.

VII. Related Work

In response to the widespread popularity of smartphones, much attention has been paid in studying application security and user privacy related issues. Extensive security analysis on smartphone apps has been carried out and can be categorized into *permission analysis*, *static analysis*, and *dynamic analysis* [3], [5], [6], [10], [13], [15], [26]–[31].

Enck et al. [10] propose Kirin, which is the first to perform inspection on Android API permissions during the app installation time to identify dangerous functionalities. Barrera et al. [5] report many applications request only a small set of permissions based on the permission analysis of top 1,100 free applications. Felt et al. [6] study over 900 applications from the Android Market and find INTERNET permission is the most frequently requested. They later propose Stowaway [9] to detect over-privilege in applications and report 10 most common unnecessary permissions.

Static analysis analyzes the code of applications to infer what can happen to users' security. For example, PiOS [26] analyzes compiled Objective-C code to identify information leaks on the iOS platform, whereas ComDroid [27] uses disassembled DEX bytecode to identify vulnerabilities in Intent

communication between applications. Enck et al. further propose the *ded* decompiler [32] to reverse Android applications to Java code for security analysis.

Some work has been done in dynamic analysis. TaintDroid [3] tracks the flow of privacy sensitive data and reports when sensitive data leaves the system via interfaces such as network connections. D2Taint [15] tracks detailed private leakage sources at runtime in multiple classes and detects leakages from any of these sources rapidly. Agarwal et al. [11] proposes ProtectMyPrivacy (PMP) which utilizes a crowdsourcing based mechanism to help users decide proper privacy settings for iOS apps. AppIntent [29] tracks the sequence of events leading to private data transmission, which helps to determine whether it is user intended or not. Zhang and Yin [30] develop Capper that can track privacy information and detect leakage at run time by inserting instrumentation code into apps. Our work takes a different viewpoint by systematically analyzing what privacy sensitive information the advertiser can collect and aggregate at run-time from multiple apps and infer the social relationship of a user. We utilizes TaintDroid as a tool to track and log the run-time privacy leakage from apps.

Finally, some work develops smartphone platform based privacy protection mechanisms, for example MockDroid [12], TISSA [14], NativeGuard [31], and AppFence [19]. Our work focuses on a different aspect of gaining systematic understanding on the social relationship inference consequences from the privacy leakage by an advertiser. Such understanding may motivate the user to adjust app usage pattern or adopt defense mechanisms to control the sensitive data leakage.

VIII. Conclusion and Future Work

Privacy leakage by smartphone apps has attracted significant research efforts in recent years. The community has proposed various defense mechanisms, from permission management, code analysis, to obfuscated data. Nevertheless, the characteristics of apps' run-time privacy leakage behavior is still not well investigated, and the consequences of such privacy leakages have not attracted much attention. This paper serves as the first step towards a comprehensive understanding of the advertiser's perspective. In particular, we seek to discover what an advertiser can infer about users' social and community relationships by combining private data from many apps. Our analysis on the run-time privacy leakage behavior of nearly 200 most popular apps from 19 categories of Google Play shows that dominant advertisers can easily gather data from many apps. We propose a privacy leakage inference framework that describes a general method for inferring users' social and community relationships. Our experimental study over one month demonstrates that an advertiser can infer 90% of users' social relationship correctly using simple heuristics. This observation is further confirmed by human mobility trace driven studies of two large scale data sets. We hope our work will eventually lead to a complete picture of the advertiser's perspective.

IX. Acknowledgement

This work is supported in part by the National Science Foundation Grants SES1450091 and CNS1217387.

References

[1] M. C. Grace, W. Zhou, X. Jiang, and A.-R. Sadeghi, "Unsafe exposure analysis of mobile in-app advertisements," in *ACM WiSec*, 2012.
[2] R. Stevens, C. Gibler, J. Crussell, J. Erickson, and H. Chen, "Investigating user privacy in android ad libraries," *IEEE MoST*, 2012.
[3] W. Enck, P. Gilbert, B.-G. Chun, L. P. Cox, J. Jung, P. McDaniel, and A. N. Sheth, "Taintdroid: an information-flow tracking system for realtime privacy monitoring on smartphones," in *OSDI*, 2010.
[4] A. P. Felt, E. Ha, S. Egelman, A. Haney, E. Chin, and D. Wagner, "Android permissions: User attention, comprehension, and behavior," in *SOUPS*, 2012.
[5] D. Barrera, H. G. Kayacik, P. C. van Oorschot, and A. Somayaji, "A methodology for empirical analysis of permission-based security models and its application to android," in *ACM CCS*, 2010.
[6] A. P. Felt, K. Greenwood, and D. Wagner, "The effectiveness of application permissions," in *WebApps*, 2011.
[7] X. Wei, L. Gomez, I. Neamtiu, and M. Faloutsos, "Profiledroid: multi-layer profiling of android applications," in *ACM MobiCom*, 2012.
[8] Lamian, "LBE Privacy Gurad," https://play.google.com /store/apps/.
[9] A. P. Felt, E. Chin, S. Hanna, D. Song, and D. Wagner, "Android permissions demystified," in *ACM CCS*, 2011.
[10] W. Enck, M. Ongtang, and P. McDaniel, "On lightweight mobile phone application certification," in *ACM CCS*, 2009.
[11] Y. Agarwal and M. Hall, "Protectmyprivacy: detecting and mitigating privacy leaks on ios devices using crowdsourcing," in *MobiSys*, 2013.
[12] A. R. Beresford, A. Rice, N. Skehin, and R. Sohan, "Mockdroid: trading privacy for application functionality on smartphones," in *HotMobile*, 2011.
[13] K. W. Y. Au, Y. F. Zhou, Z. Huang, and D. Lie, "Pscout: analyzing the android permission specification," in *ACM CCS*, 2012.
[14] Y. Zhou, X. Zhang, X. Jiang, and V. W. Freeh, "Taming information-stealing smartphone applications (on android)," in *Trust and Trustworthy Computing*, 2011.
[15] B. Gu, X. Li, G. Li, A. C. Champion, Z. Chen, F. Qin, and D. Xuan, "D2taint: Differentiated and dynamic information flow tracking on smartphones for numerous data sources," in *INFOCOM*, 2013.
[16] A. Cruz, "Farewell Facebook, and Good Riddance," http://abcnews.go.com/ABC_Univision/quitting-facebook/story?id=18668978#.UYVrQ7W854I.
[17] N. Eagle and A. S. Pentland, "CRAWDAD trace set mit/reality/blueaware (v. 2005-07-01)," Download from http://crawdad.cs.dartmouth.edu/mit/reality/blueaware, Jul. 2005.
[18] J. Bao, Y. Zheng, and M. F. Mokbel, "Location-based and preference-aware recommendation using sparse geo-social networking data," in *ACM SIGSPATIAL GIS*, 2012, pp. 199–208.
[19] P. Hornyack, S. Han, J. Jung, S. Schechter, and D. Wetherall, "These aren't the droids you're looking for: Retrofitting android to protect data from imperious applications," in *ACM CCS*, 2011.
[20] S. Thurm and Y. I. Kane, "Your Apps Are Watching You," http://online.wsj.com/article/SB10001424052748704694004576020083703574602.html.
[21] D. Knoke and S. Yang, *Social network analysis*. Sage, 2008, vol. 154.
[22] P. Hui, E. Yoneki, S. Y. Chan, and J. Crowcroft, "Distributed community detection in delay tolerant networks," in *ACM MobiArch*, 2007.
[23] H. Peng, C. Gates, B. Sarma, N. Li, Y. Qi, R. Potharaju, C. Nita-Rotaru, and I. Molloy, "Using probabilistic generative models for ranking risks of android apps," in *ACM CCS*, 2012.
[24] Google.com, "Configure access points with Google Location Service," http://support.google.com/maps/?hl=en.
[25] M. E. Newman, "Detecting community structure in networks," *EPJ B*, vol. 38, 2004.
[26] M. Egele, C. Kruegel, E. Kirda, and G. Vigna, "Pios: Detecting privacy leaks in ios applications," in *NDSS*, 2011.
[27] E. Chin, A. P. Felt, K. Greenwood, and D. Wagner, "Analyzing inter-application communication in android," in *ACM MobiSys*, 2011.
[28] L. Lu et al., "Chex: statically vetting android apps for component hijacking vulnerabilities," in *ACM CCS*, 2012.
[29] Z. Yang, M. Yang, Y. Zhang, G. Gu, P. Ning, and X. S. Wang, "Appintent: Analyzing sensitive data transmission in android for privacy leakage detection," in *ACM CCS*, 2013.
[30] M. Zhang and H. Yin, "Efficient, context-aware privacy leakage confinement for android applications without firmware modding," in *ASIACCS*, 2014.
[31] M. Sun and G. Tan, "Nativeguard: Protecting android applications from third-party native libraries," in *WiSec*, 2014.
[32] W. Enck, D. Octeau, P. McDaniel, and S. Chaudhuri, "A study of android application security," in *USENIX security*, 2011.